Solent University, Southampton

Faculty of Business, Law and Digital Technologies

**Market Analysis for the U.S Automobile Market**

Author : **Q16129351 – Victor Ifeanyi Chigbo**

Course Title : **Programming for Problem Solving – COM728**

Module Leader : **Jarutas Andritsch**

Date : **4th January 2023**

Table of Contents

[1. Overview 5](#_Toc123738163)

[2. Project Implementation 9](#_Toc123738164)

[2.1 Project structure 9](#_Toc123738165)

[2.2 CSV Data Loader Module 9](#_Toc123738166)

[2.2.1 loadDataSetasCSV Function 9](#_Toc123738167)

[2.3 DataFrame Loader Module 10](#_Toc123738168)

[2.3.1 getDatPDDF Functions 10](#_Toc123738169)

[2.4 Car Body Type Module 11](#_Toc123738170)

[2.5 Chart Analysis Module 11](#_Toc123738171)

[2.6 Project Requirement Fulfilment Modules 12](#_Toc123738172)

[2.6.1 \_1\_Car\_Record\_byID.py 12](#_Toc123738173)

[2.6.2 \_2\_Car\_Record\_Cylinder\_number.py 13](#_Toc123738174)

[2.6.3 \_3\_Car\_Spec\_Body\_type.py 13](#_Toc123738175)

[2.6.4 \_4\_Column\_Data\_Per\_Car.py 13](#_Toc123738176)

[2.6.5 \_5\_Car\_Name\_Alfa.py 14](#_Toc123738177)

[2.6.6 \_6\_Summary\_Sales.py 14](#_Toc123738178)

[2.6.7 \_7\_Top5\_Sales\_Price.py 14](#_Toc123738179)

[2.6.8 \_8\_Car\_DetailsBySelection.py 15](#_Toc123738180)

[2.6.9 \_9\_Fuel\_Consumption\_Horse\_power.py 15](#_Toc123738181)

[2.6.10 \_10\_Buying\_Behavior.py 15](#_Toc123738182)

[2.6.11 Main Module (main.py) and Text User Interface (tui.py) 15](#_Toc123738183)

[3. Recommendations 16](#_Toc123738184)

[4. Appendices 17](#_Toc123738185)

[Appendix A 17](#_Toc123738186)

Table of Figures

[Figure 1 Statistical Overview of the dataset 5](#_Toc123738977)

[Figure 2 Column Data types in the dataset 6](#_Toc123738978)

[Figure 3 Verifying the presence/absence of null values in the dataset 6](#_Toc123738979)

[Figure 4 Random Samples of 20 rows from the dataset 7](#_Toc123738980)

[Figure 5 Data format consistency 7](#_Toc123738981)

[Figure 6 Code to read dataset from csv file 9](#_Toc123738982)

[Figure 7 implementation of data format consistency 10](#_Toc123738983)

[Figure 8 data preparation to plot bar chart for car fuel system 10](#_Toc123738984)

[Figure 9 extracting unique car body type with python list comprehension and set 11](#_Toc123738985)

[Figure 10 column data list for buyer behaviors analysis 11](#_Toc123738986)

[Figure 11 retrieve car Id from list at index zero 12](#_Toc123738987)

[Figure 12 implements the zip method that displays cars by ID 13](#_Toc123738988)

[Figure 13 method with parameters 13](#_Toc123738989)

[Figure 14 Data-frame variables with user selection 14](#_Toc123738990)

[Figure 15 snippet demonstrating the implementation for 5 Sales 14](#_Toc123738991)

[Figure 16 Collection of columns of interest to display 15](#_Toc123738992)

[Figure 17 Project-Structure 1](#_Toc123738993)7

Index of Tables

Table 1: Requirement Completion 8

Table 2: Requirements and Respective Modules 12

# Overview

A Chinese company is considering entering the US automobile market by setting up a manufacturing and sales operation in the US. To help them understand which features are likely to drive sales success, the company has access to historical data on car sales in the US market. They will use this data to analyse factors that influence car purchase decisions in the US market. Some of the interesting features from the dataset that will be considered for analysis are shown in Figure 1.



Figure 1 Statistical Overview of the dataset

We want to get a statistical overview of the dataset to understand its distribution. The distribution shows that the dataset has 13 numerical columns. Some of the observations are:

1. **Price**: The mini and maximum price provides the Chinese company with the price range for automobiles in the US.
2. **Cylinder Number**: The 'cylindernumber' column shows that cars with 4 cylinders are common in the US market, as they fall within the 25th, 50th, and 75th percentiles
3. **Door Number**: The “doornumber” column shows that cars with 4 doors are more common in the US market, as they fall within the 50th to 75th percentiles in the distribution.

Figure 2 shows that the dataset has 7 columns with a data type of 'Object' (string). The dataset has a total of 205 rows and 20 columns. The column data types are displayed below;

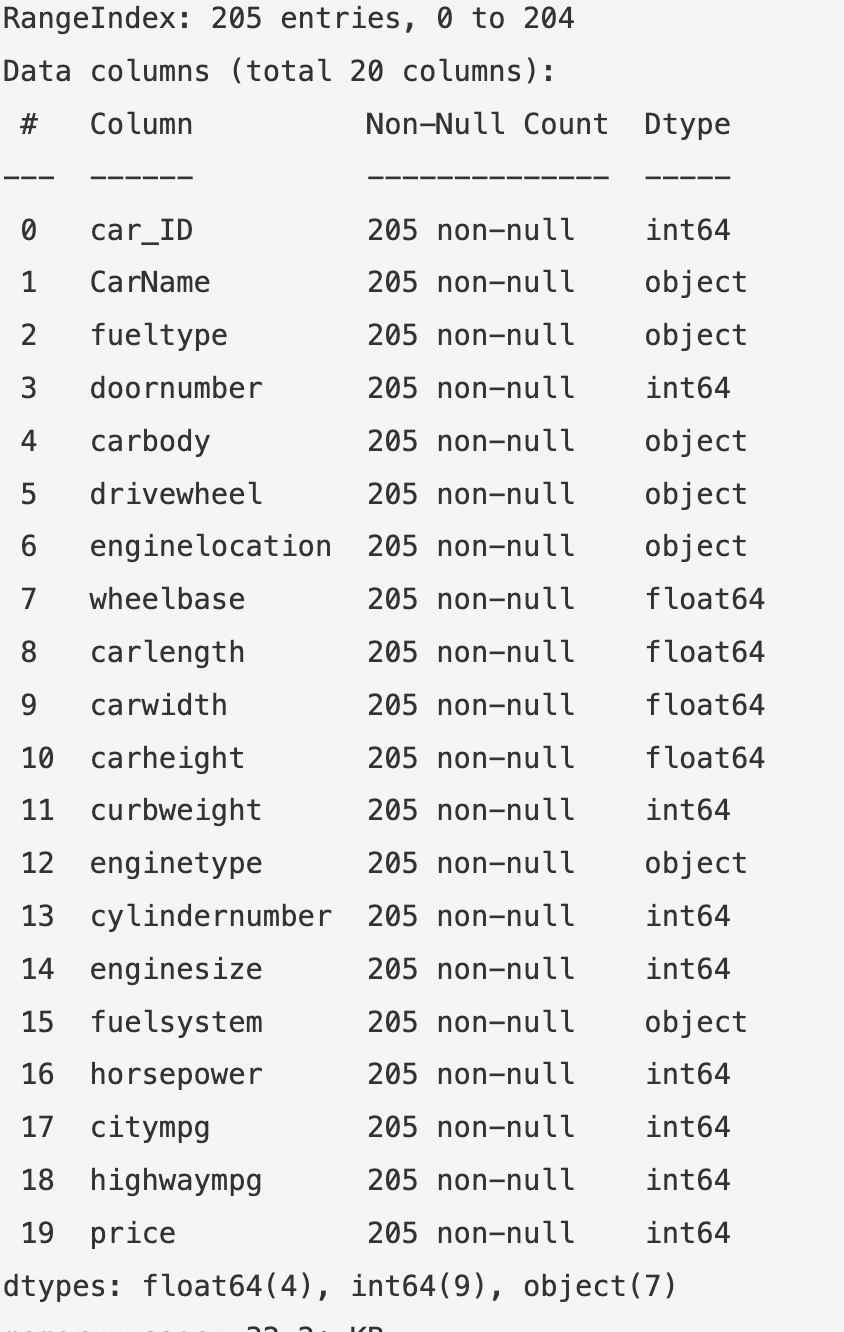


Figure 2 Column Data types in the dataset

Before analyzing the data, it is important to preprocess it. We have used two Pandas functions, 'info()' and 'describe()', to extract important information that will help us design the software artifact and answer critical questions for the Chinese automobile company.

The next step in preparing the data for our automobile market analysis will be to clean the data by correcting errors, filling missing values, and removing any irrelevant data.

Some of the data pre-processing exercise carried out in this exercise include;

1. Verify that the dataset has no null values with the “isnull().count()” pandas function on the dataset.

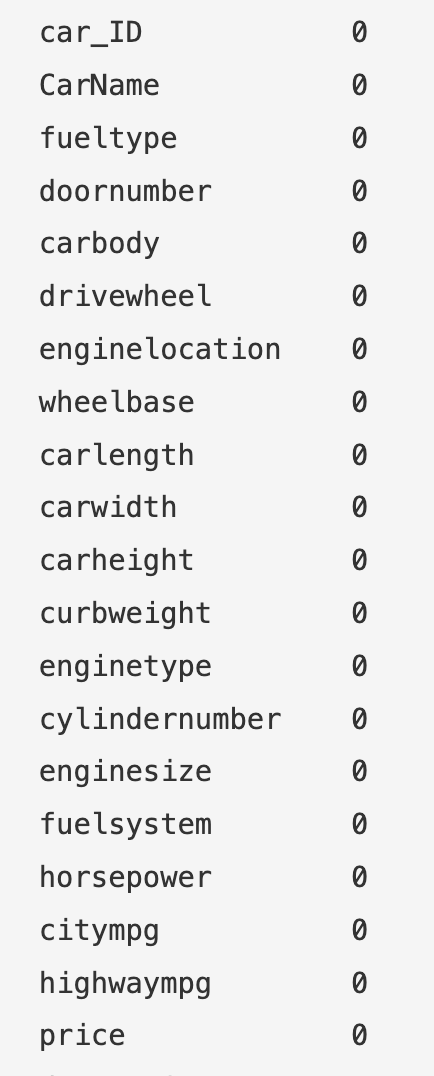


Figure 3 Verifying the presence/absence of null values in the dataset

1. For the column with “object” data type, it’s important that values in of these columns are having a consistent format. A short program was written to check column data consistency.

We will observe a random sample of the dataset to view 20 record sample. From the observation below all the values on each column are in “lower case”.

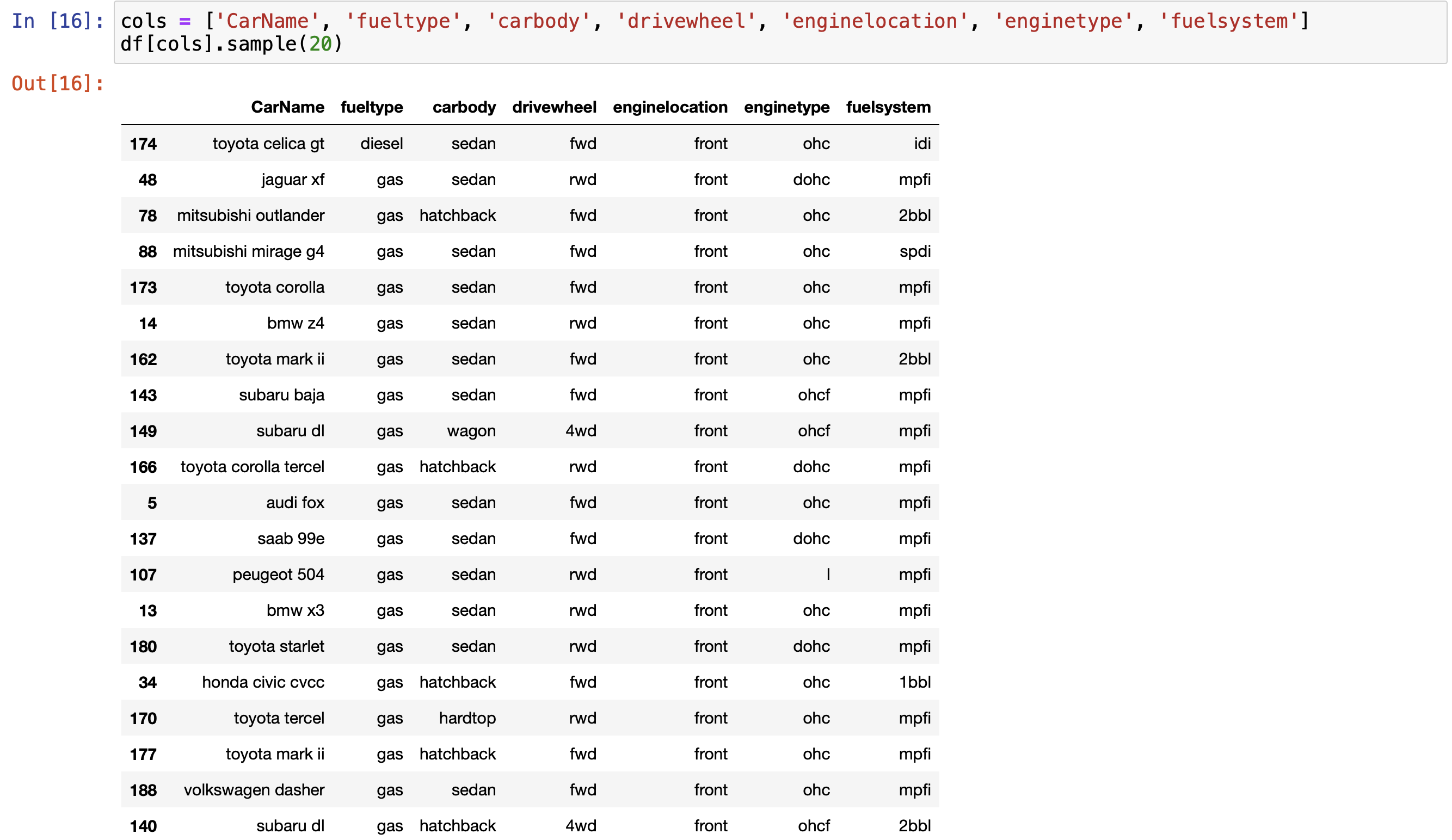


Figure 4 Random Samples of 20 rows from the dataset

A check on all columns with object data type indicates that a record in the data set starts with “upper case” under the “CarName” column. See the code snippet below.

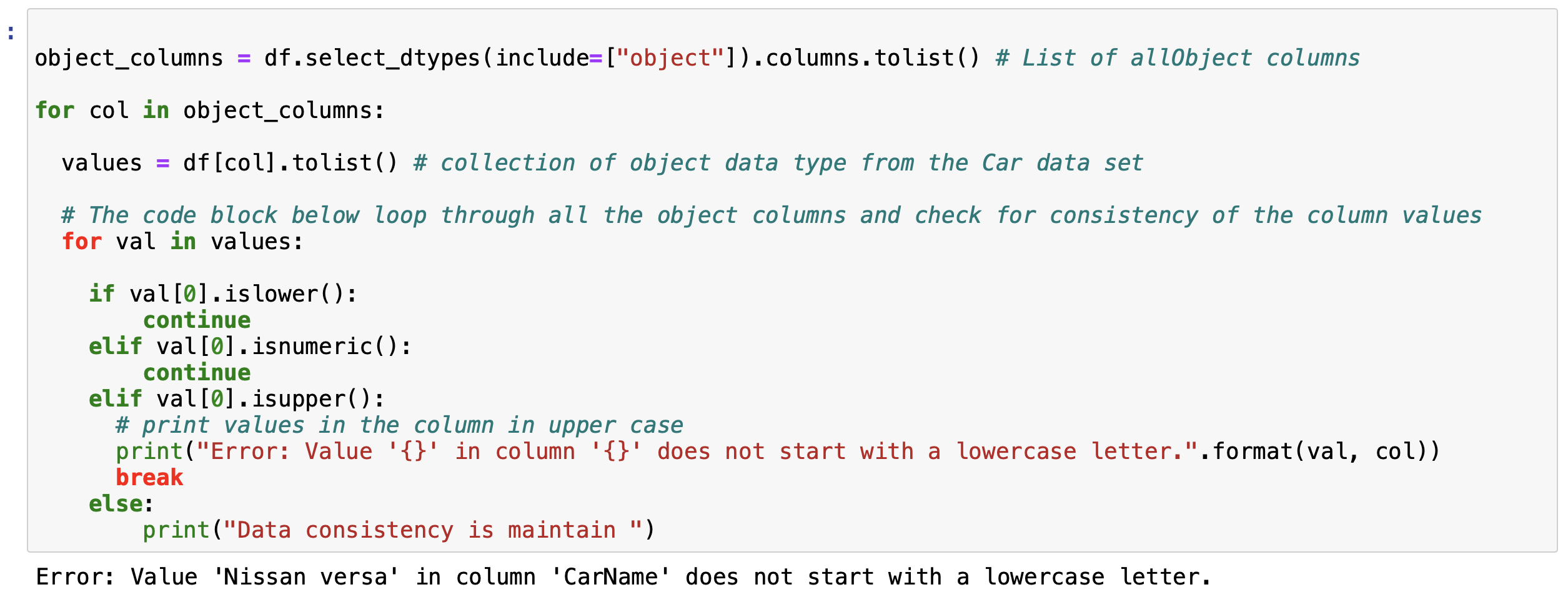
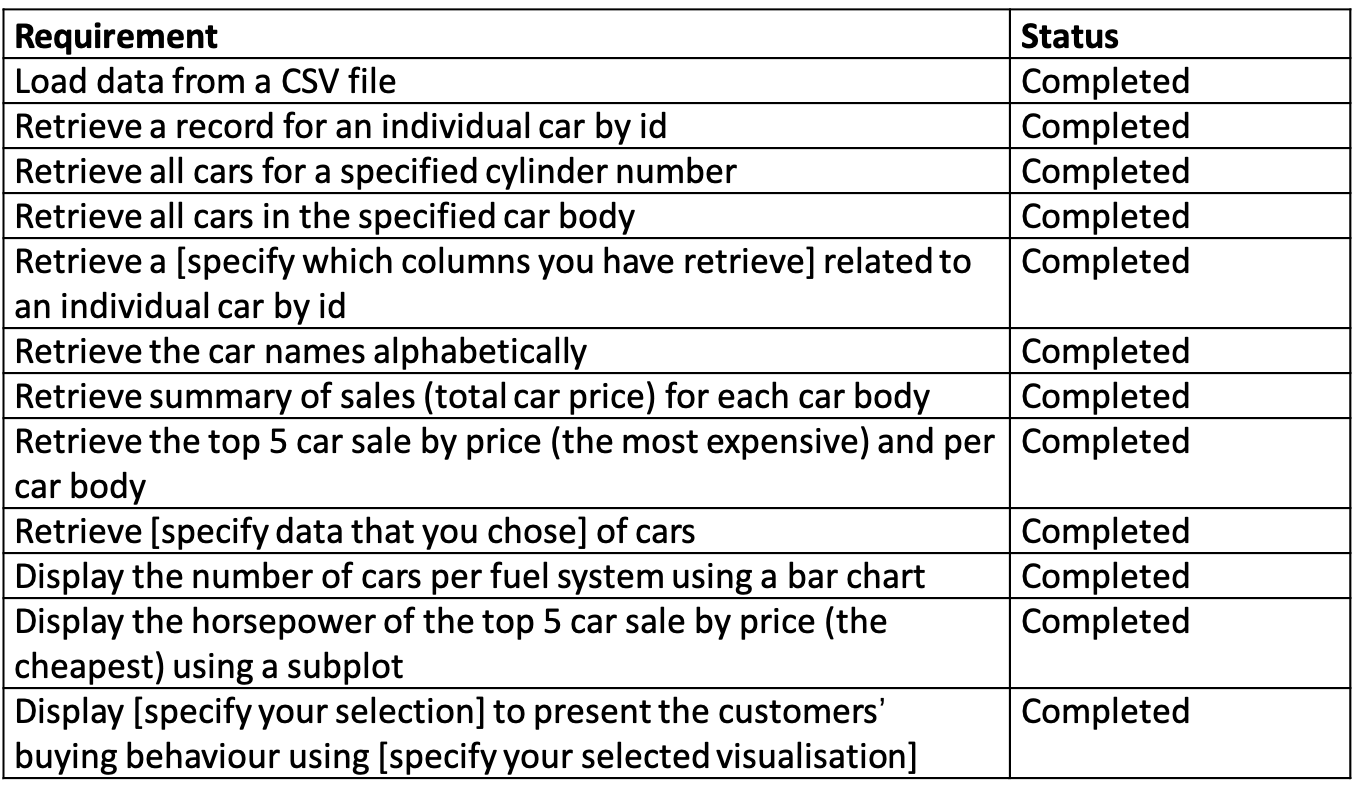


Figure 5 Data format consistency

The project completion status is outlined below.

Table 1: Requirement Completion



In the following sections, we will provide a brief overview of the structure of the project implementation design, the reasons behind our design choices, and our observations and recommendations. We will also describe how we implemented the software application.

# Project Implementation

The software implementation for this project follows a modularized development approach, with each task implemented as a standalone source-code. This design approach was chosen for the maintainability, reusability, testability, and flexibility of the software application. The naming convention used in this project was chosen to avoid ambiguity and was based on the specific requirements for each task. The names for the modules, variables, and functions in this project were carefully chosen to avoid confusion.

#### 2.1 Project structure

The software has 16 modules, 15 of which are Python files and 1 of which is a Jupyter notebook file. The 10 Python modules fulfil the requirements of the project, while the other 5 are independent modules containing functions used to process data in the initial 10 modules. The main module is a Jupyter notebook file with an interactive menu guiding the user through the software. A schematic representation of the project structure is shown in Appendix A. We will analyze the 5 independent modules that are used to prepare functions and variables for the project requirements.

#### 2.2 CSV Data Loader Module

This module, called “loaddata.py”, loads the data from a CSV file for use in the project. It uses the Python built-in library for working with CSV files, reads the file, separates the file header (columns) and records, and appends the output to two variables: “df\_header” for column names and “df\_data” for the records in each row of the file. This module includes a function called “loadDataSetasCSV”.

##### 2.2.1 loadDataSetasCSV Function

The function “loadDataSetasCSV” implements the logic for extraction of column headers and row records from the csv data set for the analysis.

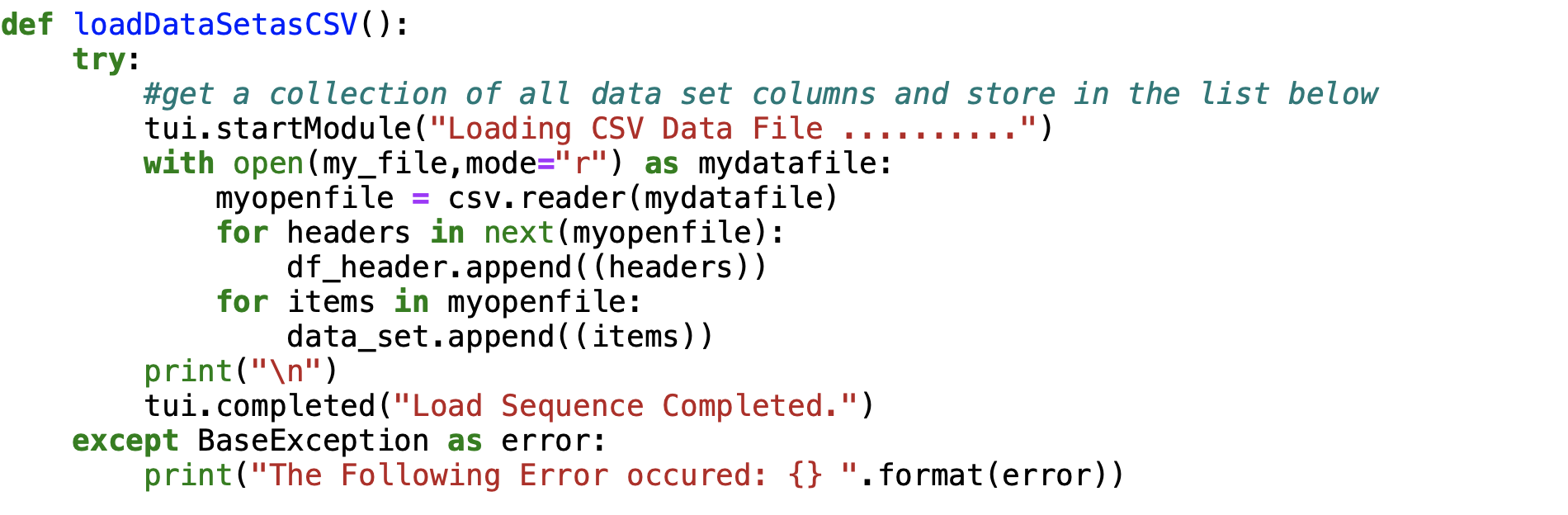


Figure 6 Code to read dataset from csv file

#### 2.3 DataFrame Loader Module

The 'getDatPDDF.py' module includes 5 functions, which will be described in the following subsections. This module creates the Pandas data-frame from the csv file. The details of the five functions in this module are outlined in the following subsections.

##### 2.3.1 getDatPDDF Functions

The functions in this module include;

1. “getCarasPDDF” function implements the read\_csv function. It reads the csv file and converts the dataset to a python data-frame.
2. “getDataColumns” function gets the list of columns from the Pandas data-frame created by the “getCarasPDDF” function. This function returns a list of the columns from the data-frame that will be used later in the project.
3. “getCarNames” implements the fix to the observation in section 1, where the data format for the “CarName” column is not consistent. The snippet below highligths the fix

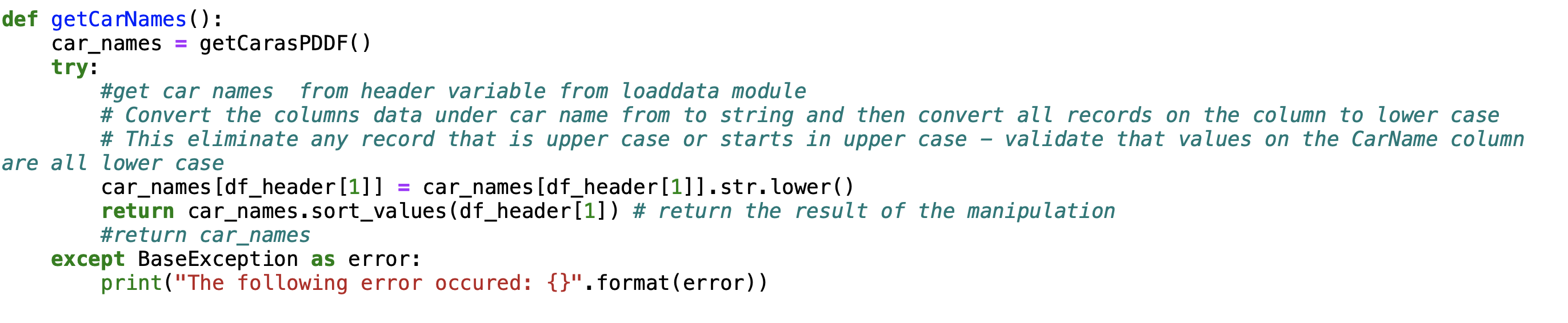


Figure 7 implementation of data format consistency

1. “getChartVariables” function groups the dataset by car fuel system and sums the total number of cars in each fuel system category. This data set is used to plot a bar chart that fulfils the requirement of the project outlined in Table 1.

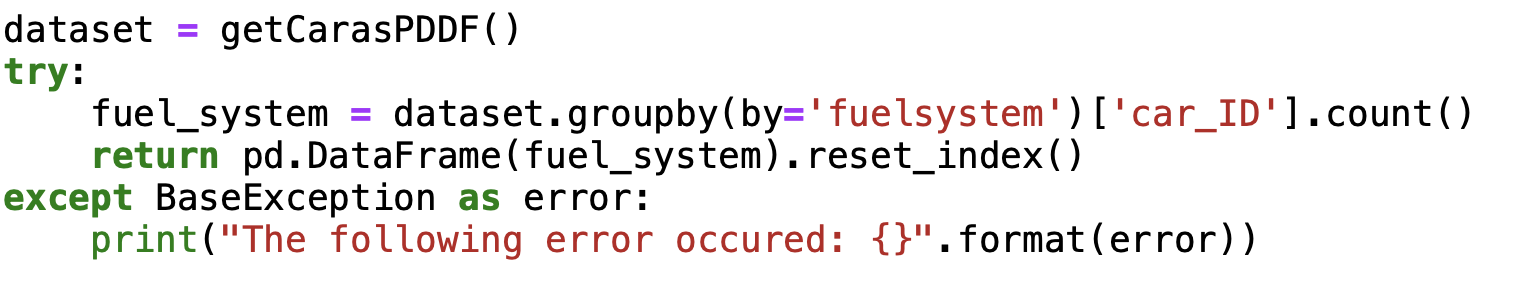


Figure 8 data preparation to plot bar chart for car fuel system

1. “getHorsePower” function groups the dataset by horse power from the Pandas data-frame used to fulfil requirement 11 in Table 1, using a subplot.

#### 2.4 Car Body Type Module

The module “getBodytype.py” implements a function “getUniqueBodyType” that extracts unique car body type from the car records data set extracted in section 2.2 (data\_set variable).

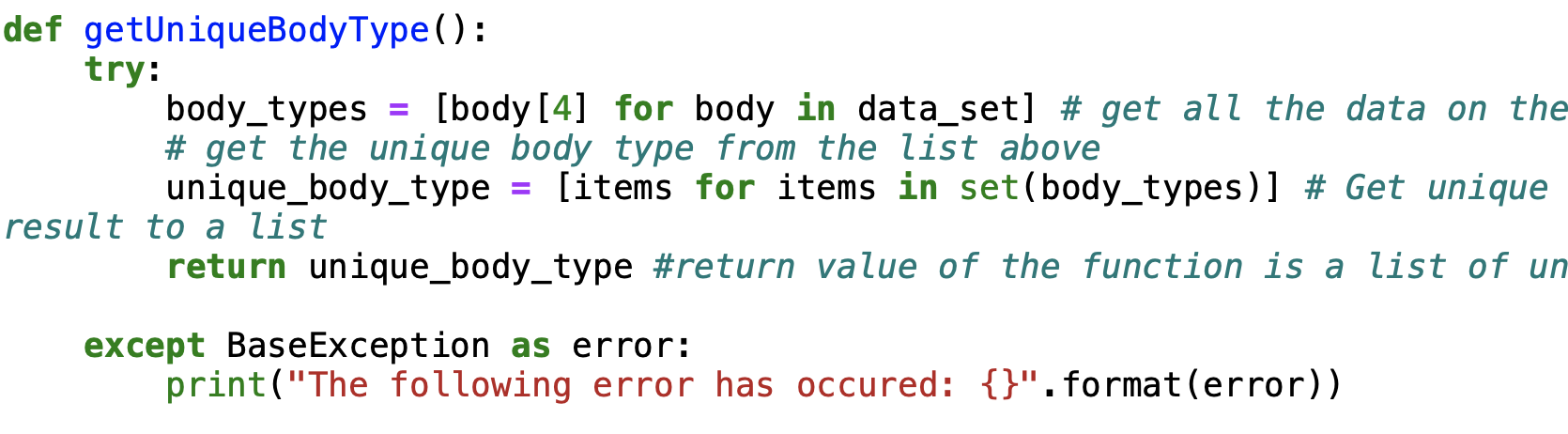


Figure 9 extracting unique car body type with python list comprehension and set

#### 2.5 Chart Analysis Module

The “chart\_analysis.py” module implements the functions that will extract datasets used to implement the charts required to view the buying behaviour of customers in the US market. It consists of two functions;

1. “getColumnofInterest” function is used to store the column data sets of interest in the “df4\_analysis” data-frame. The data-frame values will be used for analysing the buying behaviour of customers in the US market. This is highlighted in the code snippet below;

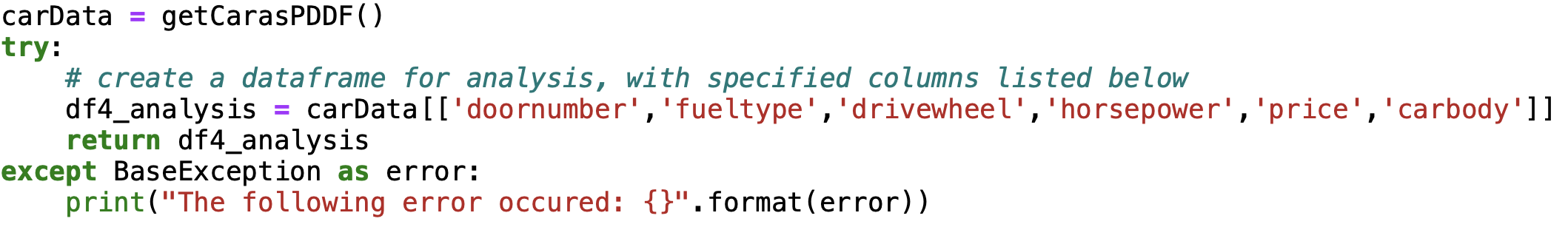


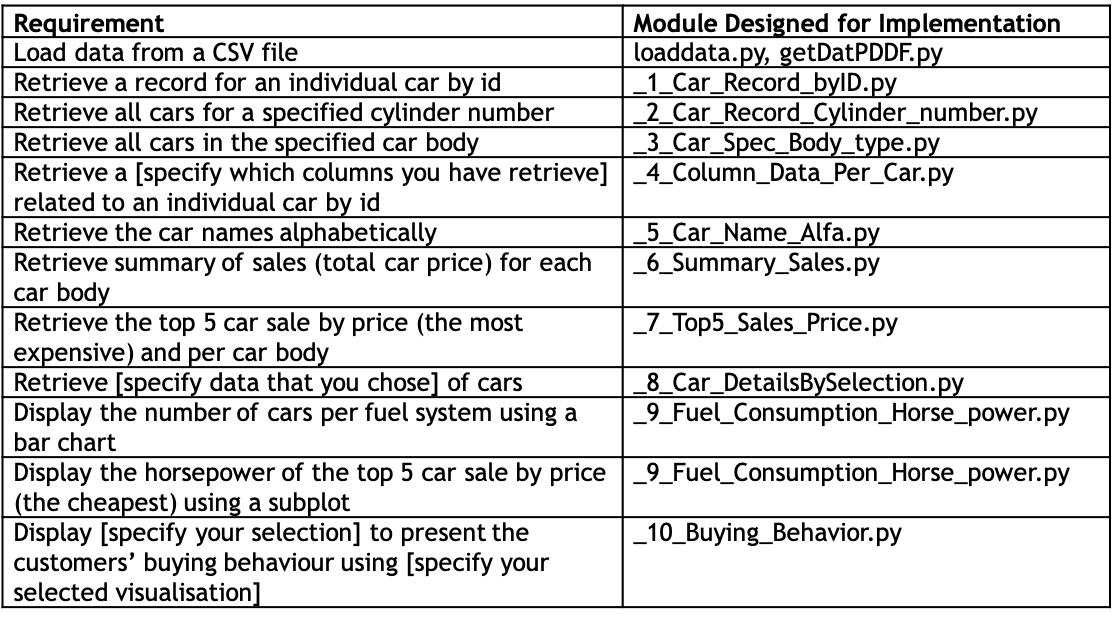
Figure 10 column data list for buyer behaviors analysis

1. “getChartofMostSalesbyDoor” function is used to extract data used to plot a bar chart showing the distribution of sales by door type in the car data set.

#### 2.6 Project Requirement Fulfilment Modules

The project implementation consists of 10 Python modules that fulfil the requirements of the project, as mentioned in section 2.1, the sections below summarizes the implementations. see Table 2 for modules designed for each requirement.

Table 2: Requirements and Modules



#### 2.6.1 \_1\_Car\_Record\_byID.py

This module consists of two functions – “getCarbyID” and “displayCrDataQuery”. Operations in the functions are;

1. “getCarbyID”, function formats the car id row index for the data-set to start at index 0 and returns id details of each car record. E.g if car\_id 1, row index for car\_id = 0

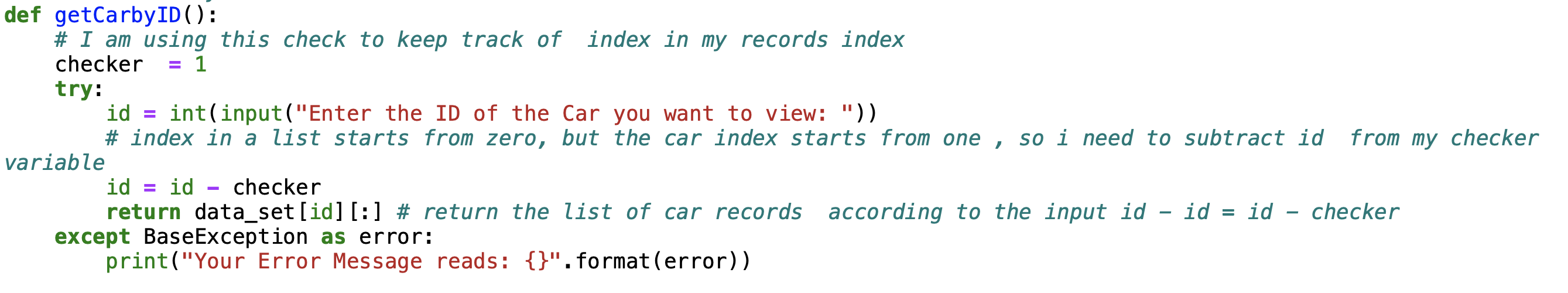


Figure 11 retrieve car Id from list at index zero

1. “displayCarDataQuery” function is used to retrieve records of individual cars by ID described above. See code snippet;

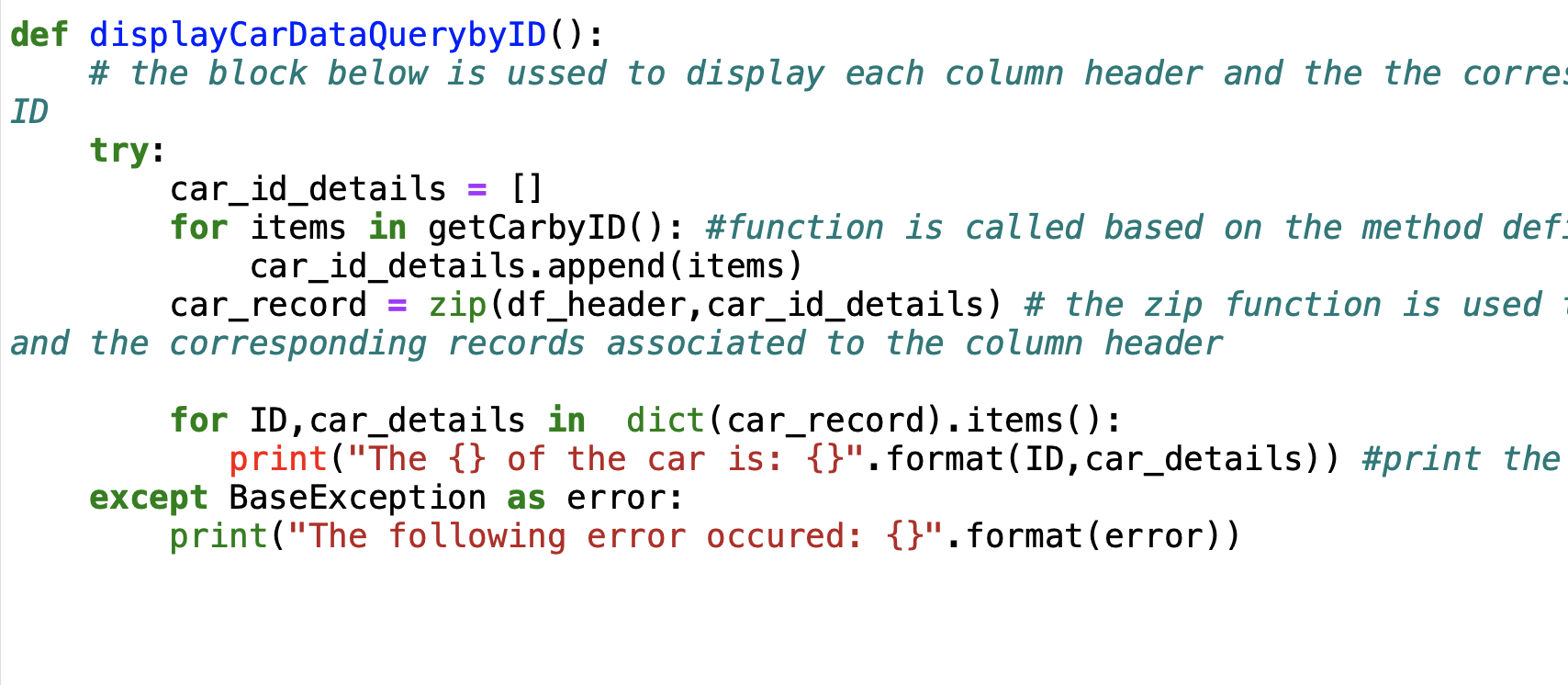


Figure 12 implements the zip method that displays cars by ID

#### 2.6.2 \_2\_Car\_Record\_Cylinder\_number.py

This module implements 2 functions “chooseCylindertoDisplay” and “getCarbyCylinderNumber”. The first function will;

1. Takes the data-frame and a cylinder value as a parameter and displays matching records from the data-frame. The cylinder number is pass to this function by “getCarByCylinderNumber) function. See snippet below;

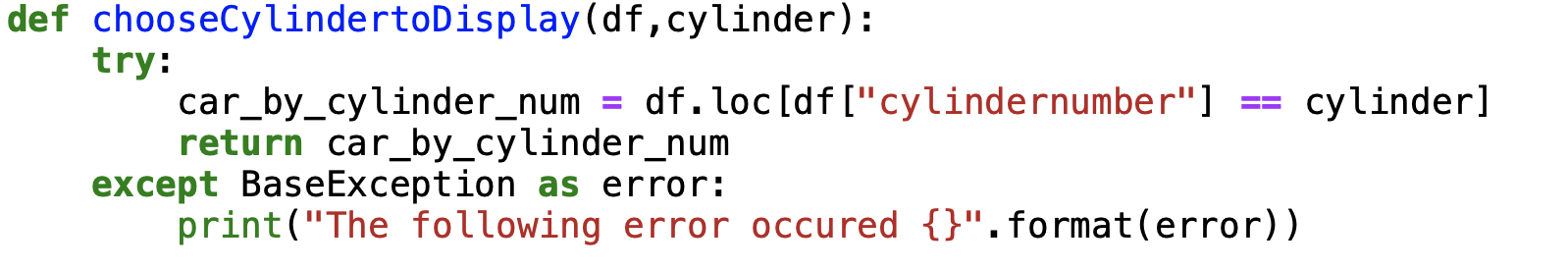


Figure 13 method with parameters

#### 2.6.3 \_3\_Car\_Spec\_Body\_type.py

The module gets a list of unique car body from “getUniqueBodyType” in section 2.4. The function “retrieveCarBody” then

1. Accepts the user’s choice of body-type
2. Displays the output of the body-type using the loc function and the data-frame parameter in the function

#### 2.6.4 \_4\_Column\_Data\_Per\_Car.py

The module implements its requirement by;

1. Giving the user a menu option to select 5 columns they are interested in viewing
2. Displays the details the of the columns by car ID select by the user by creating a new data-frame with the user’s input

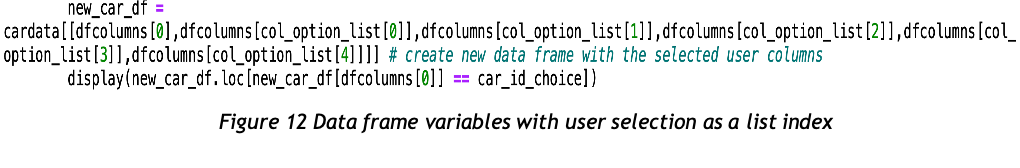


Figure 14 Data-frame variables with user selection

#### 2.6.5 \_5\_Car\_Name\_Alfa.py

This module calls the “getCarNames” function from section 1- figure-5. It converts all the car names in the column to lower case for consistency and displays the names in alphabetical order. See snippet in section 2.3.1

#### 2.6.6 \_6\_Summary\_Sales.py

The module;

1. Imports a unique sorted list of the car body, and retrieves the data on the price column of the data-set
2. Provides the user with an option to select the car body of interest from a menu it creates using the “unique sorted list”
3. Outputs the total sale per car body type

#### 2.6.7 \_7\_Top5\_Sales\_Price.py

The module;

1. Implements a function that expects a data-frame as a parameter.
2. Accepts choice of car body as input from the user using a text menu
3. Displays sales summary of 5 most-expensive cars per body selected by the user

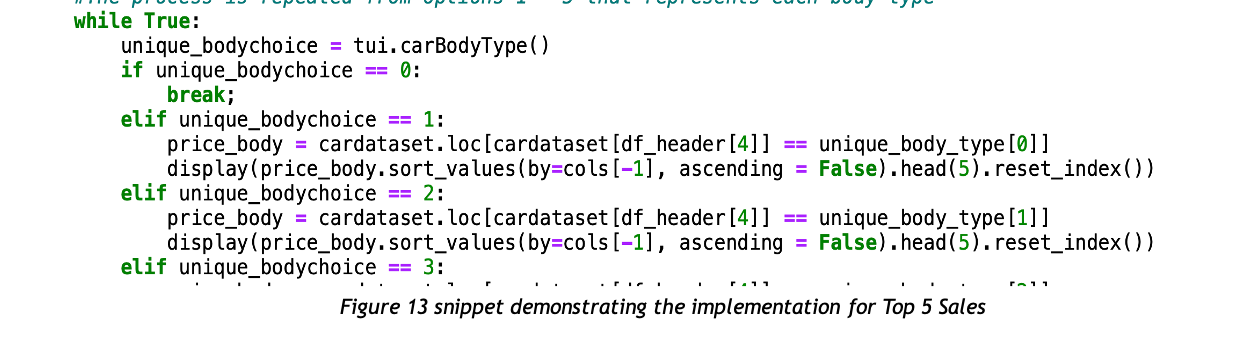


Figure 15 snippet demonstrating the implementation for 5 Sales

#### 2.6.8 \_8\_Car\_DetailsBySelection.py

The module implements a function “getSpecColumnsbyCarID” with the following tasks;

1. The function expects a data-frame as parameter.
2. A list variable “view\_feature\_list” accepts the list of columns the user is interested in.
3. The list variable is used to create a new data-frame with specified list of columns the user wants displayed.

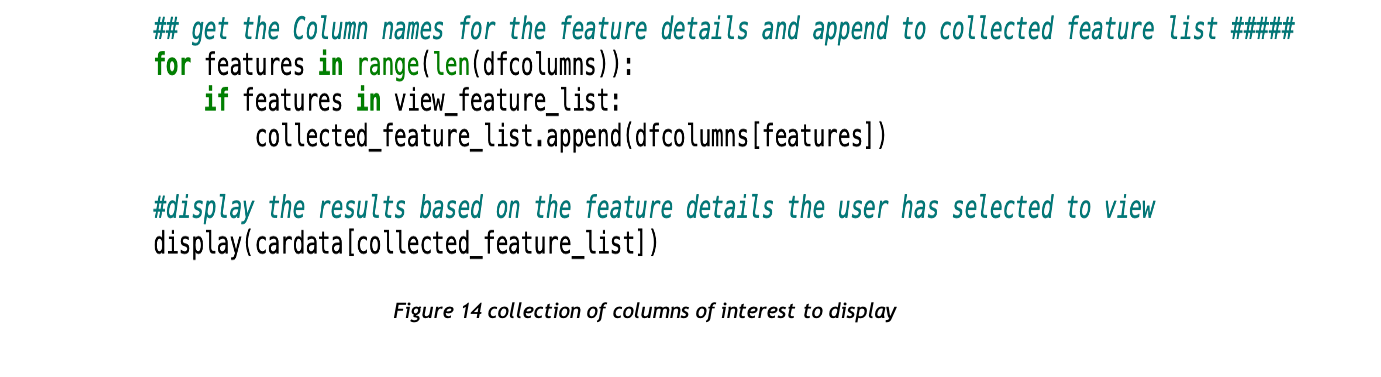


Figure 16 Collection of columns of interest to display

#### 2.6.9 \_9\_Fuel\_Consumption\_Horse\_power.py

This module implements the subplots for requirement 9 and 10 of the project using. It plots;

1. A bar-chart displaying the number of cars per fuel system
2. A line-plot for horse-power of the top 5 cheapest cars

#### 2.6.10 \_10\_Buying\_Behavior.py

In this module columns of interest for the analysis were extracted from the data-set (see section-2.5) to;

1. Plot a pie chart showing percentage of sales generated per car body type
2. Bar chart Showing the total sales generated per car body type
3. Total sales generated per car door type

#### 2.6.11 Main Module (main.py) and Text User Interface (tui.py)

The main module and TUI are the entry point for a user when they run the application. The main module imports all the modules outlined for the implementation of the project requirement and the “loaddata.py” and “getDatPDDF.py” module.

The TUI consists of 8 functions that gives the user a text interface for convenient navigation of the application.

# Recommendations

It appears that sedans and cars with 4 doors are more popular and successful in the US market, based on the data available. The Chinese company should consider these trends when deciding which types of vehicles to produce and sell in the US market in order to be competitive.

To give a more comprehensive summary of the software development, it may be necessary to increase the word count for the project documentation.

# 

# Appendices

#### 4.1 Appendix A

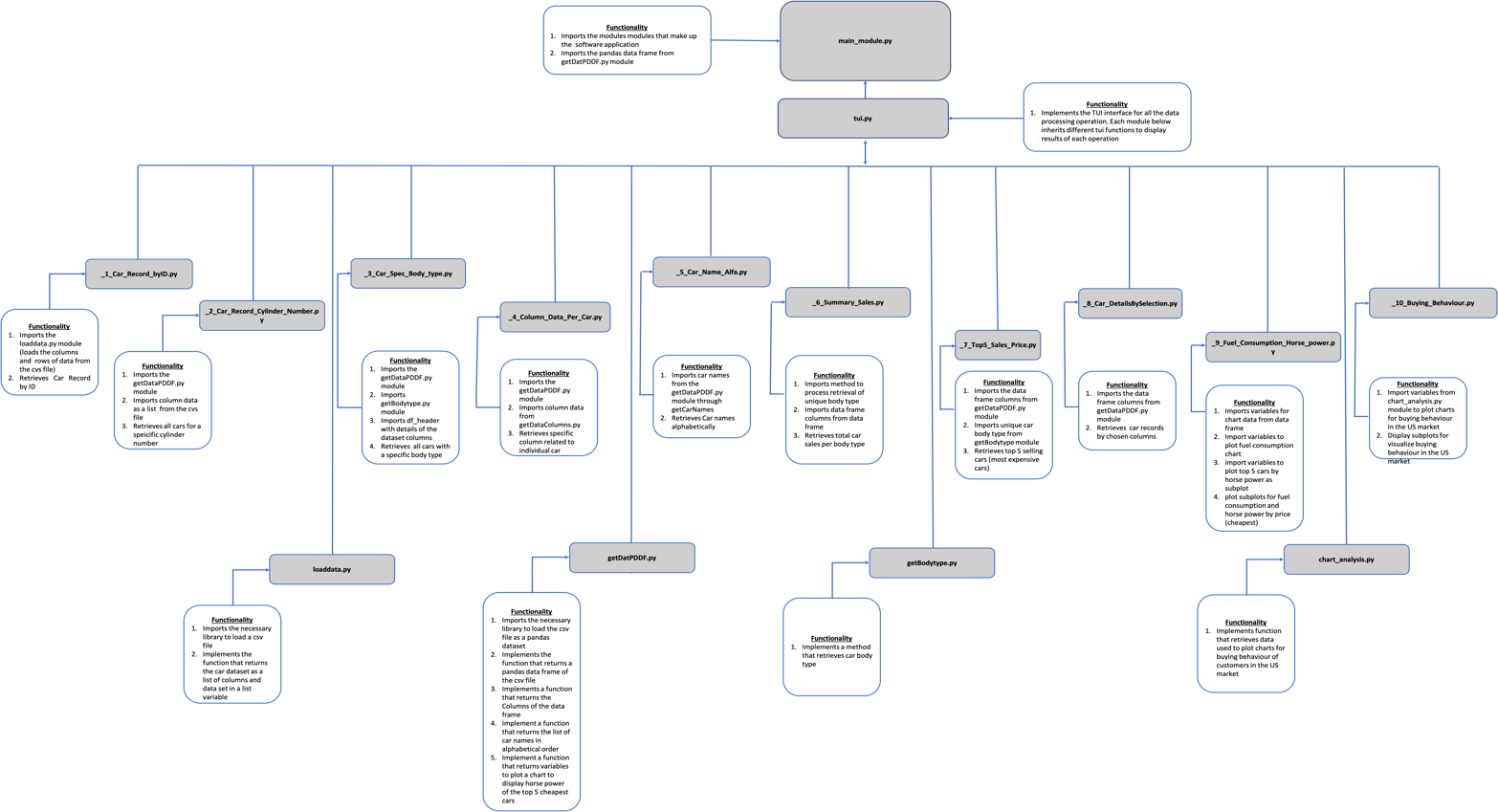


Figure 17 Project-Structure